

Control de Tráfico de Redes de Comunicación de Sistemas Distribuidos de Control

Flavia E. Felicioni^{§†} y Sergio J. Junco[§]

*§ Departamento de Electrónica, Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario*

*† Consejo Nacional de Investigaciones Científicas y Técnicas - CONICET
{flaviaf, sjunco}@fceia.unr.edu.ar*

Resumen— En primera instancia, el presente trabajo propone evaluar el tráfico de una red de comunicación a través de su “utilización” U y controlarlo dentro de un rango aceptable. En un segundo momento, y en el contexto de una red soporte de un sistema de control distribuido (NCS), se sugiere emplear información sobre la evolución de la planta para hacer este control de tráfico mediante la omisión selectiva del envío de paquetes de datos de realimentación. La idea base es que si la red se congestiona, mejor que perder datos de manera errática es hacerlo de manera selectiva y controlada, lo que permite garantizar la estabilidad y cierta performance del lazo cerrado. El método aprovecha la arquitectura distribuida e interconectada del NCS para ejecutar un conjunto de algoritmos que decide que paquetes sensor-controlador omitir. Con controladores event-driven esta omisión implica automáticamente la reducción del número de paquetes controlador-actuador. Se presentan simulaciones de un caso de estudio que permiten apreciar la eficacia del método.

Palabras Clave— Sistemas distribuidos de control, redes de comunicación, TCP/UDP, control de tráfico.

I. INTRODUCCIÓN

Los lazos de realimentación de los sistemas distribuidos de control (NCSs, por Networked Control Systems) se cierran sobre una red de comunicación digital cuyos nodos intercambian datos de sensores, controlador y actuadores vía protocolos de comunicación de tiempo real. Esta tecnología facilita la automatización con modularidad y descentralización del control, integración de diagnósticos, mantenimiento fácil y rápido, bajos costos, y mejora de la confiabilidad. En contrapartida, ciertas características de la red de comunicación -confiabilidad y ancho de banda limitados, cuantificación de datos, escalabilidad, retardos y pérdidas de mensajes- hacen que un diseño con métodos estándar de control digital no cumpla en la práctica sus hipótesis de intercambio de datos con restricciones duras de tiempo real, con el consiguiente deterioro de la calidad de control (QoC) en lazo cerrado (medida en términos de estabilidad e índices de desempeño).

Una manera típica de evitar estos problemas es sobredimensionar la red. Para ello debe considerarse tanto el tráfico periódico típico de los sistemas de control bajo operación normal, como situaciones de sobrecarga, generadas por la aparición de señales aperiódicas (p. ej., alarmas) o por la demanda de acceso a la red de otras aplicaciones que la comparten. La elección de esta alternativa conlleva un mayor costo económico para obtener (aproximadamente) la misma QoC de cada NCS que utiliza la red. Un enfoque alternativo -elegido en este trabajo- es tratar a la red como un recurso compartido y optimizar su utilización.

Hoy en día hay un creciente interés en el desarrollo de soluciones industriales Ethernet como alternativa a las redes de campo (fieldbuses). Estas redes, aunque más baratas y en ocasiones más flexibles, usan mecanismos de comunicación basados en TCP/UDP-IP, que desde el punto de vista de una aplicación NCS tienen una calidad de servicio (QoS) inferior medida con parámetros como retardo entrada-salida, jitter, ordenamiento de paquetes, entre otros (Soucek y Sauter, 2004). En estas redes, el tráfico aparece como el factor que más influencia la evolución de dichos parámetros de QoS. Por ejemplo, con una red congestionada, el uso del protocolo UDP puede ocasionar la pérdida de algunos de los paquetes enviados entre los diferentes nodos. El protocolo TCP, en cambio, puede provocar el reenvío múltiple de paquetes, aumentando la congestión y provocando entre otros el incremento del valor del tráfico y con ello de los retardos que sufren las transmisiones. En ambos casos se deteriora la QoC.

La mayoría de los trabajos que abordan el problema de los NCS lo hacen sin tomar acciones sobre la red. La tratan como una fuente de retardos constantes/variables y/o de pérdida de paquetes y analizan su influencia en la QoC. Existen trabajos donde se estudia la estabilidad y la performance frente a los mencionados fenómenos y otros donde sus efectos son compensados, por medio de: 1) cambios en la ley de control (Nilsson, 1988), (Tipsuwan y Chow, 2004), 2) compensación de pérdida de paquetes en el actuador, ZOH, o acción nula (Imer et al., 2006), 3) reducción de la utilización del recurso, usando estimadores para evitar el envío de datos (Yook et al., 2000), cambiando prioridades (Xia et al., 2006) o el período de muestreo (y del controlador).

En cambio, el presente trabajo propone evaluar la influencia de la red cuantificando su tráfico a través del parámetro denominado utilización de la red U (network utilization) mencionado en (Branicky et al., 2002), para luego controlar que su valor se mantenga dentro de valores aceptables. A tal fin, sobre la arquitectura de nodos distribuidos e interconectados del NCS, se ejecuta un conjunto de algoritmos que actúa rápidamente cuando existe un determinado nivel de congestión. La política para resolver el problema consiste en la omisión selectiva del envío de datos de mediciones (paquetes sensor-controlador), lo cual lleva implícito el beneficio de una reducción de los paquetes controlador-actuador, ya que los controladores seleccionados son de tipo event-driven. La omisión de paquetes se decide en base a información sobre la evolución de la planta.

La Sección II resume resultados previos relacionados con el tráfico de la red. La Sección III plantea el problema, el cual se resuelve en la Sección IV. La Sección V presenta un caso de estudio acompañado de simulaciones que permiten apreciar la eficacia del método propuesto. Finalmente, la Sección VI presenta las conclusiones y sugiere la continuidad del trabajo.

II. RESULTADOS PREVIOS

A continuación se analizan críticamente resultados previos afines al problema aquí tratado.

En (Xia et al., 2006) se introdujo un *scheduler* que maneja el tráfico de la red variando el período y la prioridad de cada tarea/lazo. El período de muestreo se reescala según el error entre su valor de referencia y el real usando para ello la tasa de pérdida de *deadline* M , en una extensión de la propuesta de (Lu et al., 2002) para manejar las tareas en una CPU compartida. La variación de la prioridad depende del índice IAE del NCS, una medida de la QoC. La ejecución del *scheduler* se centraliza en un nodo con un período T_s , siendo cada controlador responsable de calcular *on-line* el índice IAE de su planta y enviarlo a dicho nodo. La demanda de mensajes extra (intercambios entre controladores y *scheduler*) disminuye la eficacia teórica del método. La aplicabilidad se ve limitada por la hipótesis de relación lineal o cuadrática entre el costo y el período de muestreo usada para su reescalado, tomada de (Cervin et al., 2002). Además, una eventual simultaneidad del aumento del período y la disminución de la prioridad de un dado lazo de control (de acuerdo a los valores de M y del IAE) podría llevar a una degradación muy grande del desempeño del lazo en su próxima evaluación, con la consiguiente necesidad de aumentar su prioridad, i.e., una oscilación del algoritmo de control de tráfico. La elección del IAE como medida de performance podría crear conflictos en caso de haber distintos NCS en la misma red. Deberían usarse índices normalizados respecto al rango de excursión de cada salida para poder comparar sus valores. Por otra parte, como el IAE es un valor acumulado, se plantea un problema para lazos evaluados desde distintos tiempos iniciales.

También en base a la dinámica del sistema controlado y la utilización global, en (Velazco et al., 2004) se

presentó una estrategia dinámica de asignación de red disponible para cada lazo. Nuevamente, la disminución del requerimiento de ancho de banda se hace variando el período de muestreo de cada lazo, calculado localmente en el mismo procesador que ejecuta cada algoritmo de control. El período de muestreo se varía de manera continua y no lineal con respecto a la norma de las variables de estado. Acorde con el enfoque clásico, el período de cálculo debe cambiarse tanto en el controlador como en el sensor. El segundo cambio implica un aumento del tráfico de la red, ya que requiere el envío de una señal al sensor notificándole el nuevo valor del período de muestreo. Además, la solución requiere que los sensores puedan variar su período de forma continua. Asimismo, la elección de la norma de las variables de estado para medir y comparar la evolución de los NCS tiene problemas similares a los del índice IAE usado en (Xia et al., 2006).

Apuntando a una reducción directa de la utilización, los siguientes trabajos proponen utilizar sensores “inteligentes” que envíen sus mediciones sólo cuando las salidas superen ciertos umbrales.

La técnica asincrónica (LCS) “Level Crossing Sampling” (Kofman y Braslavsky, 2006) propone muestrear cruces de nivel de las salidas con sensores *event-driven*, i.e., que sólo muestreen cuando la salida de la planta difiera de la última muestra en un valor fijo h . El sensor transmite luego sólo un bit para indicar que la nueva muestra difiere de la anterior en $+h$ o en $-h$.

La cantidad de bits enviados se reduce fuertemente, pero no puede controlarse el tiempo mínimo entre dos bits sucesivos, lo que aumentaría la utilización de la red si una salida estuviera cruzando muy rápidamente los niveles de cuantificación en un transitorio.

En (Otanez et al., 2002a) se presenta una técnica de evaluación de resultados similares a la anterior. En cada sensor *clock-driven* se evalúa si el error entre la salida actual y la previamente enviada entra dentro de una deadband, similar al nivel h . Si entra, la medición no se envía, y viceversa. La cantidad de bits enviados es superior respecto a la propuesta anterior (en vez de 1 bit se envía una medición completa), pero se garantiza que el tiempo mínimo entre dos envíos sucesivos es el período de muestreo del sensor.

Como las propuestas solo logran estabilidad acotada (estabilidad práctica), puede haber oscilaciones finales en las salidas, como indican sus autores. No obstante, estas oscilaciones, inherentes a las estrategias de control, podrían reducirse o eliminarse con el envío al controlador de valores intermedios de la salida. En base a esta observación, el presente trabajo propone agregar al manejo periódico *clock-driven* estándar de los sensores, una detección de cruces de nivel de ciertas salidas del sistema establecidas como relevantes. Esta detección, efectuada en nodos dedicados, permitirá controlar la utilización de la red por cada nodo y la utilización global, y promover envíos complementarios para reducir o eliminar las oscilaciones mencionadas.

La coincidencia -frecuente en la práctica- de evoluciones lentas y de poca excursión de algunos lazos

(requiriendo el envío de pocas mediciones) y rápidas y amplias de otros (requiere el envío de más mediciones) favorece claramente el éxito de esta propuesta.

Como ventaja a las dos ultimas estrategias anteriores, cuyo principal objetivo es reducir la utilización que hace cada uno de los sistemas conectados al recurso red, con la consecuente reducción de la utilización global, nuestra propuesta plantea esta reducción sólo en casos de congestión, aprovechando la red para el envío de paquetes que mejoran la QoC de los sistemas cuando la utilización tiene un valor aceptable.

III. FORMULACIÓN DEL PROBLEMA

A. Modelo del Sistema de Control Distribuido

Se considera un NCS de estructura **Directa** (Tipsuwan y Chow, 2003), i.e., controlador, sensores y actuadores están ubicados en diferentes lugares, intercambiando datos via la red de comunicaciones.

En relación con el manejo de las señales de cada NCS se asume (Zhang et al., 2001): a) muestreo periódico de la salida de la planta; b) controlador y actuador manejados por los eventos a su entrada.

Bajo estas hipótesis, la propuesta de omisión selectiva del envío de datos desde los sensores a cada controlador, implica la reducción simultanea de los paquetes controlador-actuador. En consecuencia, cada actuador accede a las acciones de control de manera intermitente. De las propuestas a ejecutar cuando el actuador no recibe un paquete a tiempo, aplicar “control cero” (Imer et al., 2006) o aplicar “el último control disponible”, se elige aquí la segunda opción debido a que, en nuestra solución, la ausencia de una nueva acción de control en el actuador se corresponde con que la variación de la salida entre instantes de muestreo es acotada y por tanto el último control disponible no diferirá demasiado del *hipotético* actual.

B. Mensajes y paquetes

Transmitir en paquete único (single-packet) significa agrupar los mensajes de varios sensores a un mismo controlador en un único paquete de red y transmitirlos al mismo tiempo. Lo análogo vale para el caso de un controlador transmitiendo hacia varios actuadores. Lo contrario se denomina transmisión de múltiples paquetes (Zhang et al., 2001).

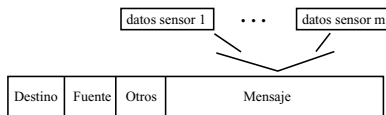


Figura 1: Sistema multi-salidas. Único paquete.

Por razones de eficiencia, en el caso de Ethernet se usará transmisión en paquete único (Zhang et al., 2001), Fig. 1, ya que tienen un máximo de 1500 bytes.

Cuando los sensores (actuadores) del sistema MIMO estén alejados espacialmente o los horizontes de tiempo de cada variable medida (acción de control) sean diferentes, se enviarán múltiples paquetes. En nomenclatura de (Nilsson, 1999), habrá múltiples nodos sensores y actuadores. En tal caso, las salidas de la

planta y entradas de control podrán no estar disponibles al mismo tiempo como se asume en control digital.

En este trabajo se propone, cuando sea posible y lógico, agrupar en un único paquete tanto las señales de sensado y como las de control.

C. Tráfico de la red

Para la transmisión de los paquetes de sensado y control requeridos en control digital, típicamente tareas periódicas, se *utiliza* la red según:

$$u_i = (T_{med} + T_{act}) * F_i \quad (1)$$

donde u_i es la utilización que hace el sistema i -ésimo, T_{med} y T_{act} son los tiempos de transmisión sensores-controlador y controlador-actuador, respectivamente, y F_i es la frecuencia de estos envíos. La sumatoria de todas las u_i da la utilización global de la red U . Si bien a los objetos de *schedulability*, U debe ser inferior a 1, en (Otanez et al., 2002b) se mostró que su valor para *Ethernet* con UDP de 10 y 100 Mb debería ser inferior a 0.25 y 0.37, respectivamente.

Si bien el tiempo de transmisión nominal para enviar cada B bits de datos sobre un canal de D bps es fijo y vale B/D segundos (Ishii y Francis, 2002), el tiempo real de envío de un paquete depende del protocolo usado para la comunicación entre planta y actuador.

En este contexto, aunque la arquitectura presentada en la sección siguiente puede ser implementada tanto en redes de campo, Token Passing Bus (ControlNet) o CAN (DeviceNet), aquí se considera una red Ethernet (CSMA/CD). La utilización (1) es sólo una estimación a-priori, por ello su valor es exacto sólo en redes de campo que garantizan los envíos.

En redes Ethernet se deben distinguir dos casos, según se use el protocolo UDP (User Datagram Protocol) o el TCP. En el primero, con los mensajes enviados según se pueda y sin ninguna garantía de recepción (política **Best-effort**), la estima (1) pierde precisión cuando alguno de los paquetes (sensor-controlador o controlador-actuador) no llega a destino. Típicamente dicha estimación para UDP tendrá un error en exceso en relación con la real utilización del canal.

En el caso de TCP, el cálculo efectivo del tiempo de transmisión debe considerar las posibles retransmisiones múltiples en una red congestionada hasta la recepción del acuse de recibo. Esta cantidad de retransmisiones es variable y depende del valor asignado al RTO (Retransmission Timeout). El mismo debería ser como máximo igual al período de muestreo del NCS para evitar la retransmisión de un dato viejo al momento que hay disponible uno nuevo (especificación de *deadline* de tiempo real duro). Una posibilidad es estimar el tiempo de transmisión promedio en condiciones de congestión, es decir cantidad de bits multiplicado por cantidad de retransmisiones promedio.

El protocolo de transporte de paquetes preferido en redes Ethernet es UDP, ya que se prefiere la pérdida de datos de las aplicaciones de control a posibles retransmisiones que no satisfagan sus *deadlines* y congestionen aún más la red (Soucek and Sauter, 2004).

Las tareas periódicas de los NCSs podrían compartir la red Ethernet, gracias a su flexibilidad, con otras aplicaciones no periódicas como información de monitoreo, diagnóstico, etc.. En ese caso la utilización no será calculable con (1), pudiendo adoptarse su expresión más general en (Lian et al., 2001) o bien considerarse otras maneras de cuantificar el tráfico. Un estimador de tráfico de la red puede encontrarse en (Tipsuwan y Chow, 2004); otra opción es medir el ancho de banda disponible para la comunicación entre dos puntos, que puede calcularse, para TCP, utilizando el algoritmo propuesto en (Cao et al., 2004). Tal método no altera el tráfico de la red porque utiliza como señales de prueba los envíos desde el nodo transmisor (sensor al controlador) y la respuesta desde el receptor (señal “acknowledgement”) en vez de las invasivas técnicas que envían numerosos paquetes de prueba.

Suponiendo constancia de los tiempos de transmisión, la reducción de utilización que hace el NCS_i se consigue disminuyendo la frecuencia F_i de tales envíos. Debido a su variación, la estimación (1) se calcula en promedio.

IV. SOLUCIÓN PROPUESTA

A. Arquitectura de Nodos Seleccionadores

La Figura 2 muestra la arquitectura propuesta. Los sensores de las diversas salidas de cada planta se conectan lógicamente a “nodos seleccionadores”, conectados a su vez a la red y encargados de decidir los datos de sensores a enviar. Tecnológicamente, estos nodos pueden ser dispositivos como *switches* o *routers*, habiéndose sugerido la incorporación de estos últimos en NCSs con redes *Ethernet* para reducir o eliminar colisiones o separar el tráfico en subredes (Otanez et al., 2002b). Tales dispositivos, además de su operación normal (direccionamiento del paquete según dirección IP destino, análisis de errores, etc.), deben disponer de una capacidad de cálculo para ejecutar los algoritmos. Funcionalmente, son llaves que habilitan el paso de paquetes en función de la utilización.

Con el controlador actualizando su valor al arribar las mediciones, surge el problema de que no todas lleguen en el mismo momento (múltiples paquetes, sistema 2). En tal situación, se propone implementar en el nodo controlador un observador o estimador de estados que permita determinar, en el momento que arriba uno de los paquetes con mediciones, si es justificado esperar los restantes paquetes.

En cada nodo seleccionador, se elige una salida denominada “de selección”, que representa la evolución de la planta evaluada en dicho nodo. Si en el camino entre sensor y controlador de cada NCS existen varios nodos (*switches* o *routers*), tal evaluación se realizará únicamente en el dispositivo más cercano al sensor.

Por ejemplo, es lógico agrupar en un único paquete las mediciones de posición y velocidad en un control de posición. Seleccionando la variable “posición”, se analizarían los bits de cada paquete correspondiente en el nodo más cercano a esta medición.

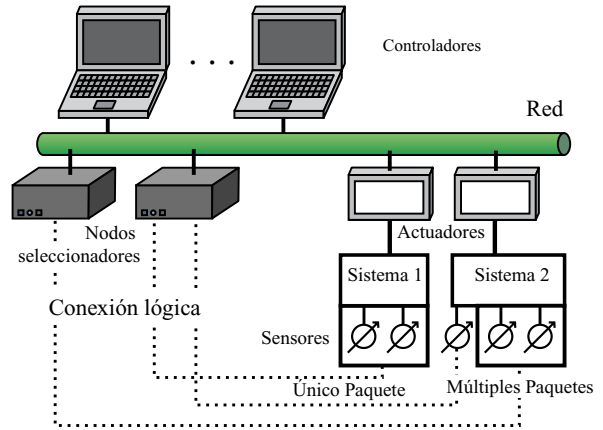


Figura 2: Arquitectura

Para cada salida de selección se ejecutará el algoritmo de sección IV-C que utiliza una técnica de detección de cruce obtenida como una variación en tiempo discreto de LCS (Kofman y Braslavsky, 2006).

B. Control de la Utilización de la Red

El tráfico de la red se cuantifica con la utilización global $U = \sum_{j=1}^R U_j$ donde U_j es el parámetro que mide los

envíos emitidos efectivamente cada cierto tiempo por cada uno de los R nodos seleccionadores. La utilización de cada nodo se calcula $U_j = \sum_{i=1}^M u_i$ con u_i dado por (1) y M es la cantidad total de salidas de selección por nodo.

La adopción de UDP permite considerar tiempos de transmisión constantes como se mencionó en sección III-C. Bajo ésta hipótesis, en lo que sigue se habla de la frecuencia F_i en vez de la utilización u_i de cada salida.

Mientras la frecuencia de muestreo de cada una de las M salidas de selección $1/h_i$ es constante, la frecuencia de los envíos de paquetes F_i variada en cada nodo se calculará realizando el siguiente promedio.

Si en el tiempo actual t_{ac} se va a enviar una medición y hasta el tiempo t_{ant} se enviaron d mediciones:

$$F_i(t_{act}) = \frac{(d+1)}{(t_{act} - t_0)} = \left(F_i(t_{ant}) + \frac{1}{(t_{act} - t_0)} \right) \frac{(t_{ant} - t_0)}{(t_{act} - t_0)} \quad (2)$$

La implementación de este control de la utilización de la red se hace según el esquema jerárquico de fig. 3. En cada capa del mismo se calculan las señales que son de referencia para la capa inmediatamente inferior.

La utilización global del sistema U se calcula, cada un tiempo T_u , en algún dispositivo conectado a la red, el cual recibe un mensaje con el valor de U_j desde cada nodo seleccionador. Por lo que T_u debe ser grande para no sobrecargar el sistema. Mientras que los cálculos de utilización de cada nodo y la frecuencia de cada NCS se realizan localmente en cada nodo.

La reducción global se logra promoviendo que cada nodo reduzca en un dado porcentaje la cantidad de paquetes enviados. Para esto, cada cierto tiempo T_k se calculan las M frecuencias de referencia F_i^* de cada salida de selección requeridas en cada i algoritmo.

Desde luego, U puede ser inferior o superior que su objetivo (referencia U^*), descongestión o congestión, respectivamente. En el primer caso, $U < U^*$, y no hay congestión ($F_i^* = 1/h_i$). En el segundo caso se debe reducir U de manera que $U \approx U^* (F_i^* \downarrow)$.

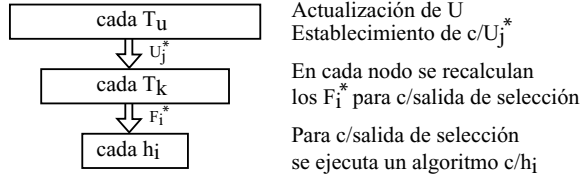


Figura 3: Escala temporal, jerarquía. Tareas en cada capa.

Suponiendo tiempos de transmisión ($T_{tx,i}$) iguales, las frecuencias de referencias en cada j -ésimo nodo se calculan en $(n+1)T_k$ según:

$$F_i^*(nT_k + T_k) = k_i(nT_k + T_k) F_i(nT_k + T_k)$$

donde $r_i(nT_k + T_k) = r_i(nT_k) + F_i(nT_k + T_k)/F_i^*(nT_k)$ y

$$k_i(nT_k + T_k) = \left(\frac{U_j^*(nT_k + T_k)}{T_{tx}} \right) / \left(\sum_{i=1}^M (r_i(nT_k + T_k) F_i(nT_k + T_k)) \right)$$

Inicialmente $k_i(0) = (U_j^*(0)/T_{tx}) / \left(\sum_{i=1}^M 1/h_i \right)$ y $r_i(0) = k_i(0)$.

La utilización total y las utilidades de referencia de cada nodo se actualizan de manera similar cada T_u .

$$U_j^*(nT_u + T_u) = K_j(nT_u + T_u) U_j(nT_u + T_u)$$

donde $g_j(nT_u + T_u) = g_u(nT_u) + U_j(nT_u + T_u)/U_j^*(nT_u)$ y

$$K_j(nT_u + T_u) = U^*(nT_u + T_u) / \left(\sum_{j=1}^R (g_j(nT_u + T_u) U_j(nT_u + T_u)) \right)$$

siendo $K_j(0) = U^*(0) / \left(T_{tx} \sum_{j=1}^R \left(\sum_{i=1}^M 1/h_i \right) \right)$ y $K_j(0) = g_j(0)$.

En general, los F_i alcanzan su F_i^* en diferentes tiempos. A través de la reactualización de las F_i^* se disminuyen las referencias de sistemas cuyas F_i hayan alcanzado su valor de referencia y amplían las referencias de los sistemas restantes, es decir, se reduce el envío de mediciones de sistemas cuyas salidas están evolucionado lentamente y se permite el aumento de la frecuencia para aquellas que lo hacen rápidamente. El éxito de este esquema depende fundamentalmente de que coexistan evoluciones transitorias de algunos sistemas con evoluciones permanentes de otros.

C. Algoritmo para cada Salida de Selección

El algoritmo recibe la referencia F_i^* y calcula (2). Según el error y teniendo en cuenta principalmente la velocidad de los eventos de cruce se abre o cierra la llave que permite el envío de mediciones por la red hacia el nodo donde se ejecuta la ley de control, Fig. 4.

Siguiendo criterios de estabilidad (sección IV-E), la frecuencia de cada NCS se limita inferiormente. O sea, F_i puede asumir ciertos valores discretos entre $1/h_{max,i}$ y $1/h_i$ (el límite superior es el período de muestreo).

El tiempo de respuesta de cada algoritmo depende de la frecuencia de transmisiones, limitada superior e inferiormente.

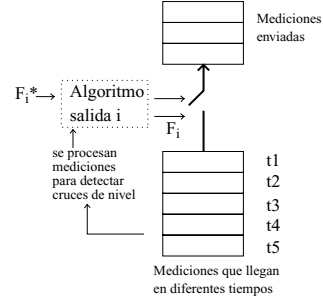


Figura 4: Esquema de comportamiento del algoritmo.

Este algoritmo de dos pasos, se ejecuta cada vez que arriba la medición de una salida de selección al nodo. En el paso 1, se verifica si hubo un cruce de nivel entre el valor actual de la salida y su valor anterior. En caso positivo y si además el nivel del cruce anterior no es igual al actual (histéresis), se decide enviar la medición. Fig. 5.

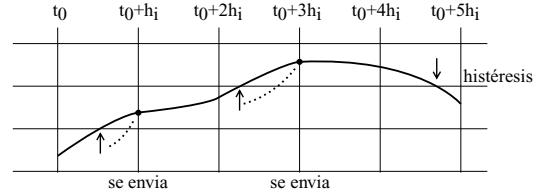


Figura 5: Evaluación de cruce de nivel de la salida.

En caso de que no haya habido un cruce se evalúan las dos posibilidades:

- de enviar la medición, resultaría $F_i \leq F_i^*$? Si la respuesta es positiva, se decide enviar.
- transcurrió el tiempo máximo h_{max} desde el último envío? Si la respuesta es positiva, se decide enviar.

En la parte 2 del algoritmo, se recalcula la frecuencia después de enviar la medición.

D. Nivel de cruce

Para el buen funcionamiento del algoritmo es la elección adecuada, en relación al período de muestreo, del valor del nivel de cruce N (o quantum), que debe ser suficientemente grande para que el tiempo de cruce de dos niveles sucesivos por la salida de selección sea igual o superior al período de muestreo. Para esto se debe tener una idea de la respuesta temporal de cada salida de selección. En este contexto, se utilizará la siguiente idea, que si bien es válida sólo para un sistema discreto con un período fijo, se extenderá aquí para nuestros propósitos.

Para un sistema LTI la respuesta temporal puede caracterizarse por el tiempo de crecimiento T_r (rise time), tiempo necesario para que la salida partiendo del 10% alcance el 90% de su valor final. Y en TD, T_r puede ser interpretado como R veces el período de muestreo. Siguiendo esta idea (Åström et al., 1997) sugieren la regla práctica $0.2 \leq \omega_b h \leq 0.6$, propuesta originalmente para la elección del período de muestreo. Esta regla surge a partir de considerar los casos extremos $R=10$ y $R=3.5$ en la siguiente expresión: $\omega_b h \approx 2.2/R$. Un valor de R superior a 10, $T_r \gg h$ se

interpreta como un sobremuestreo, mientras un valor ligeramente inferior a 3.5 resulta en $T_r \approx h$.

A partir de esto, para calcular el nivel de cruce de la salida se propone $N \geq 0.8c/R$. Si se quiere perder alguna medición durante el transitorio el valor c debe ser superior a 1, pero también se debe considerar agrandar R . Por ejemplo, $c=2$ y $R=4$ implica un valor de N superior o igual a 0.4 (niveles del 40 % del valor final), lo que resulta muy grande. En cambio con $c=1$ (cruce de niveles en cada período de muestreo) resulta $N=0.2$.

E. Cálculo De La Ley De Control, Estabilidad, QoC

La variación de frecuencia de envíos (y de actuación), se modela como un sistema discreto de período de muestreo variable $\Delta_v h_i$. Para el caso de enviar todas las salidas del sistema en un único paquete, el vector de estados evoluciona desde un tiempo t_v a t_{v+1} como sigue $x_{t_{v+1}} = \Phi(\Delta_{v+1} h_i) x_{t_v} + \Gamma(\Delta_{v+1} h_i) u_{t_v}$, donde $\Phi(\Delta_v h) = \Phi^{\Delta_v}(h)$ y $\Gamma(\Delta_v h) = \left(\sum_{r=0}^{\Delta_v} \Phi^r \right) \Gamma(h)$ (Van Loan, 1978) y $t_{v+1} - t_v = \Delta_v h_i$

En caso de múltiples paquetes se incluye el término de ruido para modelar la no recepción de todas las mediciones $x_{t_{v+1}} = \Phi(\Delta_{v+1} h_i) x_{t_v} + \Gamma(\Delta_{v+1} h_i) u_{t_v} + \eta_v$. Es decir, si se reciben mediciones desde un nodo y de otro no, se corresponderá con que la salida de selección evaluada en el segundo no cruzó su nivel por lo cual habrá un error entre su valor actual y el anterior acotado por $2N_i$.

Para garantizar la estabilidad en sistemas discretos de período variable puede ser diseñado un controlador por realimentación de estados utilizando el método de (Sala, 2005) que usa LMI (desigualdades de matrices lineales) y funciones cuadráticas de Lyapunov (función común).

La QoC del NCS se regula considerando la velocidad de evolución de la salida de selección (vía detección de cruce de nivel) y limitando el tiempo máximo de envíos.

A pesar de que la solución no considera explícitamente la pérdida errática (no controlada) de alguno de estos mensajes, si se perdiera la acción de control actualizada y la aplicación del último control disponible produjera, en el sigte. instante de muestreo, un nuevo cruce de nivel de la salida de selección, nuevamente se reintentaría corregir la acción de control. Es decir, esta supervisión de las salidas del sistema permite, frente a la presencia de algunas pérdidas esporádicas no forzadas, mantener la estabilidad.

V. CASO DE ESTUDIO

Sean R nodos, todos con M salidas de selección. De acuerdo a sus períodos de muestreo y su tiempo de transmisión, la utilización global nominal $T_{ex} \sum_{j=1}^R \left(\sum_{i=1}^M 1/h_i \right)$ es 1. Se adopta $U^*=0.2$, ya que para 10 Mb dicho valor debería ser inferior a 0.25. Entonces, inicialmente $K_1(0)=\dots=K_4(0)=0.2$, e igualmente en cada nodo, $k_1(0)=\dots=k_4(0)=0.2$.

Si el NCS 1 del nodo 1 tiene un período $h_1=0.02$, entonces $F_1^*(0)=10\text{Hz}$. Se asume que NCS 1 es una

planta con dos salidas que ingresan en un único paquete. La posición y es la salida de detección de la planta:

$$y/u_c = k_2 / (s(s + k_1)), \text{ con } k_1=12.65588 \text{ y } k_2=1.9243.$$

El control por realimentación de estados $L=[113.433 \ 6.292]$ se calcula con la versión discreta de LQR y h_1 , donde $Q=[1 \ 0; 0 \ 0]$ y $R=6 \cdot 10^{-5}$. Luego, usando el teorema 7 de (Sala, 2005) se prueba la estabilidad para períodos que varían entre h_1 y su máximo $h_{max}=0.14$. Por esto, la mínima frecuencia alcanzable es $50/7=7.14 \text{ Hz}$.

Para este controlador $\omega_0=\pi/(10 \ h_1)$, por lo que $R=2.2/\omega_0^2 h_1=7$. Eligiendo $c=2$, para reducir envíos en el transitorio, $N=0.22$, y se adopta $N=0.2$. Los niveles están centrados en 0 y equiespaciados en N . Con este valor de N , los cruces de niveles durante el transitorio ocurren en tiempos mayores a un período de muestreo.

El objetivo inicial es reducir el envío de mediciones a un 20% del nominal, $F_{nom,i}=50$ y $F_i^*=10$. En $t=0$, ingresa un escalón de referencia de posición de valor 1.1, el cual no es múltiplo entero de N . Tal entrada hubiera causado oscilaciones en torno al valor final en un control LCS con $N=0.2$. En $t=1.62$, ingresa una perturbación escalón aditiva en la salida del sistema.

Suponiendo el peor escenario posible, es decir que los otros NCSs también estén en sus evoluciones transitorias y reduzcan sus envíos según puedan, la referencia F_i^* se mantendrá en su valor constante de 10.

Para mostrar la potencialidad del algoritmo se simuló tres casos (evoluciones en Fig. 7): el NCS nominal $S1$ (negro) con frecuencia de envíos igual a la de muestreo, $F_i=1/(h_i)$; el NCS $S2$ (verde) con frecuencia reducida al valor de referencia $F_i^*=1/(5h_i)$; y el NCS $S3$ (azul) que usa el algoritmo aquí presentado. Se aprecia que durante la respuesta al escalón la evolución del $S3$ se encuentra entre las evoluciones obtenidas con $S1$ (nominal $1/h_i$) y la deteriorada del $S2$.

Si bien la QoC es regulada implícitamente al usar el algoritmo, se calcula *off-line* otro índice J (costo LQR discreto) para comparar la performance de los NCSs. El valor de J , durante la respuesta al escalón ($t < 1.5$), para $S1$ es de 0.1373, y para $S2$ se degrada con respecto al de $S1$ en un 42.5%. Utilizando el algoritmo, $S3$, J se degrada sólo un 6.4%.

La fig. 8 muestra para el $S3$ la diferencia temporal entre dos eventos de cruce sucesivos para cada tiempo. Se aprecia que la frecuencia en el transitorio no se reduce demasiado, pero el tiempo entre eventos es superior o igual al período de muestreo h_1 . Se ve que una vez que la salida de $S3$ cruza el último nivel ($t > 0.2$) el tiempo de muestreo es aumentado al máximo h_{max} hasta conseguir que F_i se reduzca a F_i^* . Cuando esta se alcanza, la frecuencia de envíos es $1/(5h_1)$.

Cuando ingresa la perturbación escalón en $t=1.62$ el algoritmo detecta el cruce de nivel e inmediatamente se envía la medición, generando que $S3$ actúe de manera similar a $S1$. Para $S2$, la perturbación no es detectada hasta $t=1.7$ que es múltiplo del período de muestreo.

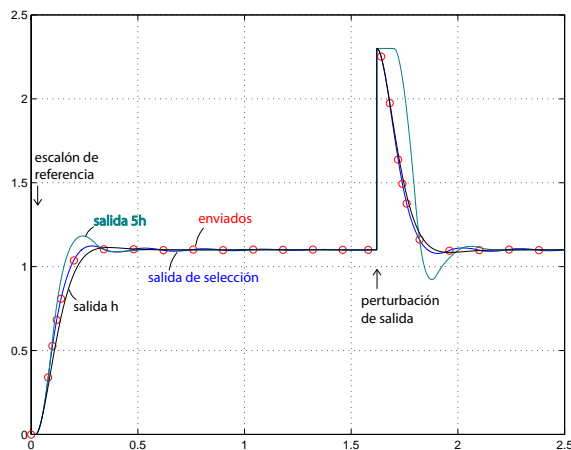


Figura 7: Respuesta al escalón y a perturbación.

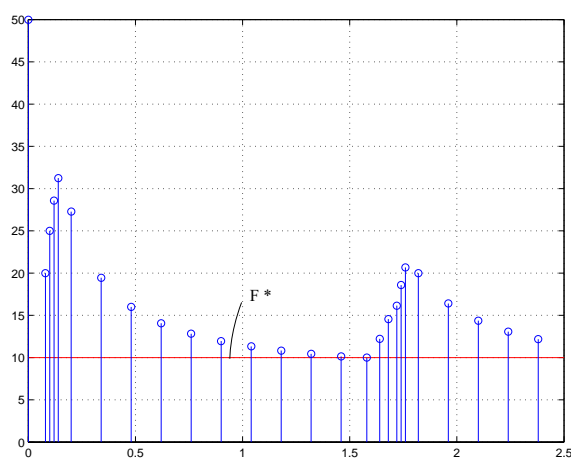


Figura 8: Evolución de la frecuencia de envíos.

La velocidad con la que la frecuencia alcanza su referencia depende principalmente del valor de h_{max} y de la existencia de eventos de cruce. La aparición de muchos eventos de cruce hace más grande el tiempo en el cual el valor real alcanza la referencia.

VI. CONCLUSIONES

Se presentó una arquitectura jerárquica para controlar la utilización del ancho de banda de una red mediante la omisión selectiva del envío de paquetes de datos. El nivel más alto evalúa cada cierto tiempo una utilización global U y establece referencias para la utilización U_j de cada nodo seleccionador físicamente distribuido. El nivel intermedio regula la utilización U_j de cada nodo contra su referencia, para lo cual establece frecuencias de referencia para cada sistema de control que transmite desde ese nodo. El nivel inferior implementa la política antes mencionada sobre nodos que transmiten la información de las salidas de sensores, lo que aporta un doble efecto de descongestión de la red al usarse controladores manejados por eventos a su entrada, que no recalculan sus salidas si no reciben nuevos datos de mediciones.

El uso de esta estrategia en un caso de estudio mostró la reducción parcial de envíos durante transitorios relevantes, con bajo deterioro del índice de desempeño. Con evolución cercana al estado de régimen la

frecuencia de envíos disminuye sustantivamente hasta alcanzar la frecuencia de referencia. La aparición de una perturbación se detecta rápidamente y permite la acción de control, lo que no ocurriría si se aumentara el período de muestreo. También se mostró que no hay oscilaciones finales y que el tiempo mínimo entre dos envíos no puede ser inferior al período mínimo de muestreo.

Como continuación, se prevé distribuir totalmente el algoritmo de reducción de frecuencias, a partir de encontrar (o adaptar) métodos de cuantificación del tráfico de una red, no invasivos de rápida convergencia, como, por ejemplo, medir el *ancho de banda disponible* para la comunicación entre dos puntos.

AGRADECIMIENTOS: Los autores agradecen al Profesor Julio H. Braslavsky (Centre for Complex Dynamic Systems and Control, University of Newcastle) por sus valiosas sugerencias.

REFERENCIAS

- Åström, K. J. y B. Wittenmark, "Computer-Controlled Systems", third edition. Prentice Hall (1997).
- Branicky M., M.S., Phillips, y Zhang W., "Scheduling and Feedback Co-Design for Networked Control Systems". Proc. of the 41st IEEE conf. on dec. and contr. 1211–1217 (2002).
- Cao M.L.T., G. Hasegawa y M. Murata, "Available Bandwidth Measurement via TCP Connection". in Proc. of IFIP/IEEE MMNS (2004).
- Cervin A., J. Eker, B. Bernhardsson y K.-E. Årzén, "Feedback-Feedforward Scheduling of Control Tasks", Real-Time Systems, 23(1), 25-53 (2002).
- Kofman E. y J. Braslavsky, "Level Crossing Sampling in Feedback Stabilization under Data-Rate Constraints". In Proc. of CDC'06: 4423–4428, San Diego, USA (2006).
- Imer O. C., S.Yükselb y T. Başar. "Optimal control of LTI systems over unreliable communication links". Automatica. 42: 1429 – 1439 (2006).
- Ishii H. y Francis B., "Limited data rate in control systems with networks". Springer Verlag (2002).
- Lian F., J. Moyne, y D. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet"-Contr. Syst. Mag. IEEE (2001).
- Lu C., J. Stankovic, G. Tao y S. H. Son, "Feedback Control Real-Time Scheduling: Framework, Modeling, and Algorithms" J. Real-Time Syst., Special Issue on Control-Theoretical Approaches to Real-Time Computing (2002).
- Nilsson J. "Real-time Control Systems with Delays", PhD Thesis, IT Lünd. Sweden, January (1998).
- Otanez P., Moyne J., y Tilbury D., "Using Deadbands to Reduce communication in Networked Control Systems," American Contr. Conf. (2002a).
- Otanez P., J. Parrott, J. Moyne, y D. Tilbury, "The implications of Ethernet as a control network", Proc. of the Global Powertrain Conf. (2002b).
- Sala A., "Computer control under time-varying sampling period: An LMI gridding approach", Automatica (2005).
- Soucek S. y T. Sauter, "Quality of Service Concerns in IP-Based Control Systems", IEEE Trans. on Ind. Electr., 51(6), (2004).
- Tipsuwan Y. y M.-Y Chow, "Control methodologies in networked control systems". Contr. Eng. Pract. 11:1099–1111. (2003).
- Tipsuwan, Y. y M.-Y. Chow, "Gain Scheduler Middleware: A Methodology to Enable Existing Controllers for Networked Control and Teleoperation—Part I: Networked Control" IEEE Trans. On Ind. Electr., 51(6): 1218 – 1227 (2004).

XII Reunión de Trabajo en Procesamiento de la Información y Control, 16 al 18 de octubre de 2007

- Van Loan C., "Computing integrals involving the matrix exponential", IEEE Trans. Automat. Contr., 23:395-404, (1978).
- Velasco M., J. M. Fuertes, P. Martí y S. A. Brandt, "A Control Approach to Bandwidth Management in Networked Control Systems", Proc. 30th IEEE IECON, Korea, (2004).
- Xia F., L. Liu, S. Li, y Y. Sun, "Integrated Feedback Scheduling of Networked Control Systems", J. of Dynamics and Cont., Disc. Time and Impulsive Systems (2006).
- Yook J., D. Tilbury, H. S. Wong, y N. Soparkar. "Trading computations for bandwidth: State estimators for reduced communication in distributed control systems". Proc. of the Japan-USA Symp. on Flexible Automation. (2000).
- Zhang, W., M. Branicky y S. Phillips. "Stability of networked control Systems". IEEE Contr. Sist.. Mag.: 84-99 (2001).